

# Feature preserving Delaunay mesh generation from 3D multi-material images

Dobrina Boltcheva, Mariette Yvinec, Jean-Daniel Boissonnat

GEOMETRICA – INRIA Sophia Antipolis, France  
firstname.lastname@sophia.inria.fr

---

## Abstract

*Generating realistic geometric models from 3D segmented images is an important task in many biomedical applications. Segmented 3D images impose particular challenges for meshing algorithms because they contain multi-material junctions forming features such as surface patches, edges and corners. The resulting meshes should preserve these features to ensure the visual quality and the mechanical soundness of the models. We present a feature preserving Delaunay refinement algorithm which can be used to generate high-quality tetrahedral meshes from segmented images. The idea is to explicitly sample corners and edges from the input image and to constrain the Delaunay refinement algorithm to preserve these features in addition to the surface patches. Our experimental results on segmented medical images have shown that, within a few seconds, the algorithm outputs a tetrahedral mesh in which each material is represented as a consistent submesh without gaps and overlaps. The optimization property of the Delaunay triangulation makes these meshes suitable for the purpose of realistic visualization or finite element simulations.*

Categories and Subject Descriptors (according to ACM CCS): Computational Geometry and Object Modeling [I.3.5]: Curve, surface, solid, and object representations—Modeling packages

---

## 1. Introduction

Advanced medical applications, like numerical simulations of physical or physiological processes, frequently require meshed models of human organs rather than grey-level MRI or CT-scan images. Extracting geometric representation from a medical image usually supposes that the image has been *segmented* into several regions of interest made of different materials. During this *segmentation*, every voxel is assigned a label so that connected voxels with the same label represent an anatomical structure. This paper focuses on the next step and presents a method that, given a segmented 3D image, meshes simultaneously a set of regions with different labels. The resulting meshes are either surface meshes approximating the boundaries of anatomical structures or volume meshes of these structures.

Building meshed models from multi-material images is very challenging. Indeed, in dimension 3, the different materials are often organized in intricate geometric configurations and they present three kinds of multi-material *junctions*. First, the *2-junctions* are surface patches defined as

the interface between 2 different regions of the image. Secondly, the *1-junctions* are edges defined as the intersection of 3 or more image regions. Thirdly, the *0-junctions* are corner vertices appearing at the meeting point of 4 or more regions.

All these multi-material junctions form features in the input and are expected to be preserved and accurately represented in the output mesh for many applications. Besides, if the image regions are meshed separately, it is not obvious to guarantee that the resulting meshes have conforming boundaries without gaps or overlaps along their common junctions. In this paper, we present a feature preserving Delaunay refinement algorithm which addresses this issue. The method simultaneously meshes all image regions and builds a unique high-quality tetrahedral mesh where each anatomical structure is represented by a submesh. In addition, the method is constrained to preserve the *0-junctions* between regions and to reconstruct accurately the *1-junctions* as a set of consecutive Delaunay edges and the *2-junctions* as surface patches composed of Delaunay facets.

### 1.1. Related work

There are only a few surface mesh generation methods which deal with multi-material 3D images. The most popular are extensions of the Marching Cubes algorithm handling multi-material junction configurations. In an early work, Hege et al. [HSSZ97] proposed the first generalization of this algorithm for segmented data. Their look-up table supports up to three different materials meeting at one grid cell. More recently, Wu and Sullivan [WS03] proposed another extension dealing with up to eight different labels per cell.

A well-known issue with MC methods is that they produce very dense meshes presenting staircase artefacts and that they are composed of numerous distorted triangles. Meshes generated from a typical segmented medical image contain an average of  $10^6$  triangles. Thus, the initially generated meshes are usually smoothed (using Laplacian based filters) and decimated to obtain a mesh with reasonable size and quality. However, concurrently controlling the accuracy of this mesh filtering and preserving its topological properties (such as the absence of self-intersections) turns out to be a difficult and time consuming task. In addition, the usual methods do not guarantee that the resulting meshes are conforming at the 0- and 1-junctions where ambiguous topological configurations exist.

Less popular, but much simpler are Dual Contouring methods which operate on the dual grid obtained by shifting the hexahedral image grid by half a cell width in each direction. These methods have been initially proposed for binary images [Gib98, Nie04] and output quadrilateral meshes corresponding to the intervoxel boundaries. As for the MC methods, the generated meshes need smoothing and decimation in order to give acceptable tessellations in terms of quality and size. Dual contouring methods do not rely on look-up tables and their extension to multi-material images is straightforward.

In a recent work, Bertram et al. [BH05] proposed an algorithm for dual contouring of segmented medical images. The obtained non-manifold mesh is made of all quadrilateral surfels separating two voxels with different labels. It contains ambiguous topological configurations when 18- or 26-connected voxels with the same label are adjacent, i.e. when they share only one edge or one vertex. Reitingner [RBB05] proposed a surfel subdivision strategy that does not guarantee the absence of singularity. Recently, Bischoff and Kobbelt [BK06] proposed a voxel subdivision strategy that removes these *critical* edges and vertices before the extraction of the intervoxel boundaries. Both methods do not explicitly handle the 0- and 1-junctions during the mesh smoothing and decimation. Thus the resulting meshes may present gaps or overlaps along these features.

There are only a few meshing strategies which directly provide volume meshes from multi-material images. Early grid-based methods deal with tetrahedral elements that are created from the original rectilinear volume subdivision

[NF97, HK98]. More recently, an octree-based method has been proposed [ZHB07] that outputs tetrahedral or hexahedral meshes with consistent multi-material junctions. However, like the other grid-based methods, this algorithm does not offer a mesh quality control and elements of poor quality are always generated along boundaries. Quality improvement techniques are usually used as a post-processing step to deliver acceptable meshes.

A powerful strategy based on Delaunay refinement, has been proposed for meshing smooth surfaces [BO05] and volumes bounded by smooth and piecewise smooth surfaces [ORY05, RY07]. The refinement process is controlled by highly customizable quality and size criteria on triangular facets and on tetrahedra. Pons et al. [PSB\*07] have adapted this mesh generation strategy to segmented images and have shown that the sizing criteria can be material-dependent leading to high-quality volume meshes of the different materials where 2-junctions are accurately represented by connected sets of Delaunay facets. However, in this work, 0- and 1-junctions are not explicitly handled and they are not accurately represented in the output mesh. One way to circumvent this drawback is to extract the 0- and 1-junctions from the input 3D image and to constrain the Delaunay refinement to preserve them. However, it is well recognized that the presence of constrained edges where meeting surface patches form small angles, jeopardize the termination of the Delaunay refinement in 3D [She00]. Recent methods [CDR07] deal with this problem using the idea of protecting small angle regions with balls so that during the refinement no point is inserted into these protecting balls.

Very recently, Mayer et al. [MWK\*08] proposed a sampling strategy for segmented 3D images based on a dynamic particle system which explicitly samples all multi-material junctions. The resulting set of points is then meshed using a Delaunay-based algorithm which outputs high-quality surface meshes where, in particular, corner vertices are preserved and junction edges are accurately represented. However, their sampling approach relies on heavy preprocessing computations and takes between 3 to 12 hours for small medical images.

### 1.2. Our contribution

In this paper, we propose a feature (edge and corner) preserving extension of the Delaunay refinement algorithm for 3D multi-material images, introduced by Pons et al. [PSB\*07]. This extension handles 0- and 1-junctions using the mechanism of protecting balls introduced by Cheng et al. [CDR07] to deal with sharp edges when generating meshes for objects with piecewise smooth boundaries.

The basic idea is to first sample points on the 0-junctions and 1-junctions and to cover these features with balls centred at the sampled points. These balls are called *protecting balls*. The Delaunay refinement algorithm is then tuned

to use a *weighted* Delaunay triangulation, initialized with the *weighted points* corresponding to the protecting balls. The refinement algorithm then inserts points into the mesh until all simplices meet some topological and user-given quality criteria. Initially sampled points are preserved in the mesh. Therefore, 0-junctions are accurately represented and the weighting scheme constrains the refinement algorithm to preserve the representation of any 1-junction as a sequence of mesh edges joining consecutive sampled points.

In the case of Cheng et al. [CDR07, CDL07], sharp edges are part of the input and known a priori. In the present application to biomedical imaging, we propose to detect all multi-material 0- and 1-junctions into the input 3D image. More precisely, we introduce a multi-material junction extraction algorithm which detects all multi-material junctions where 3 or more materials meet. This algorithm operates on the only available input, which is the 3D multi-material image, and uses its regular hexahedral grid to extract the junctions. Furthermore, we also explicit a precise scheme to relax quality criteria for simplices incident to the 0- and 1-junctions for the Delaunay refinement algorithm. Such a relaxation of quality criteria near 0- and 1-junctions is crucial to avoid infinite loops during the refinement.

Note that, in contrast to Mayer et al. [MWK\*08], we extract and protect only 0- and 1-junctions because the 2-junctions are efficiently meshed by the Delaunay refinement algorithm according to the quality and sizing criteria. In addition, our multi-material junction extraction and protection algorithm is very fast and has little influence on the computation time of the Delaunay refinement algorithm.

The remainder of this paper is organized as follows: Section 2 gives a brief background and recalls the Delaunay refinement algorithm. Section 3 presents the detection of multi-material junctions, the protection strategy for 0- and 1-junctions and the required adaptation of quality criteria near the constrained 0- and 1-junctions. Section 4 reports some results and numerical experiments which demonstrate the effectiveness of our approach for segmented medical 3D images.

## 2. Delaunay refinement mesh generation from images

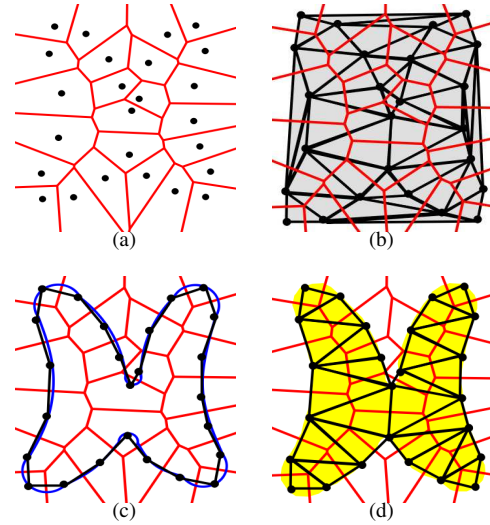
The method is related to the concept of *restricted Delaunay triangulation*, borrowed from computational geometry. In this section, we briefly recall some basic notions of this field but the interested reader should refer to [BY98] and references therein for more detail.

### 2.1. Background

The restricted Delaunay triangulation is a powerful tool for mesh generation in 3D.

Let  $E = \{p_0, \dots, p_n\}$  be a set of points in  $\mathbb{R}^3$ . The *Voronoi cell*, denoted  $Vor(p_i)$ , associated to a point  $p_i$  is the region

of  $\mathbb{R}^3$  defined by points that are closer to  $p_i$  than to all other points in  $E$ . The *Voronoi diagram* of  $E$ , denoted  $Vor(E)$ , is the partition of space induced by the Voronoi cells  $Vor(p_i)$ . The *Delaunay complex*  $Del(E)$  is defined as the geometrical dual of the Voronoi diagram. Each cell in  $Del(E)$  is the convex hull  $conv(T)$  of a subset  $T \subset E$ . The cell  $conv(T)$  belongs to  $Del(E)$  if and only if the intersection  $\cap_{p \in T} (Vor(p))$  of the Voronoi cells of points in  $T$  is non empty. If points of  $E$  are in general position, i.e. there are no 5 points lying on the same sphere, each cell in  $Del(E)$  is a simplex and the Delaunay complex is a triangulation. The fundamental property of the Delaunay triangulation is that a tetrahedron belongs to the triangulation iff its circumball does not contain any other point of  $E$  in its interior.



**Figure 1:** (a) A set of points in the plane and its Voronoi diagram. (b) The dual Delaunay triangulation. (c) The Delaunay triangulation restricted to the blue curve is plotted with a black line. (d) The Delaunay triangulation restricted to the yellow region is composed of triangles whose circum-centres are inside the region.

Let us now consider a subset  $\Omega \subset \mathbb{R}^3$  and a set of points  $E$ . We call Delaunay triangulation of  $E$  *restricted* to  $\Omega$ , and denote  $Del(E)|\Omega$ , the subcomplex of  $Del(E)$  composed of Delaunay simplices whose dual Voronoi faces intersect  $\Omega$ . Fig.1 gives an example of these notions in 2D.

In 3D, if the Delaunay triangulation is restricted to a surface  $S$  then  $Del(E)|S$  is a triangular mesh composed of all facets of  $Del(E)$  whose dual Voronoi edges intersect  $S$ . Otherwise, if  $Del(E)$  is restricted to a volume  $V$  then  $Del(E)|V$  is a tetrahedral mesh composed of tetrahedra whose circum-centres are contained in  $V$ . In both cases, under some condition on the sampling  $E$ , the restricted Delaunay triangulation forms a good approximation of the object. Indeed, under the assumption that  $E$  is a *sufficiently dense* sample of  $\Omega$ , it

has been proven that  $Del(E)|\Omega$  is homeomorphic to  $\Omega$  and that the Hausdorff distance between  $Del(E)|\Omega$  and  $\Omega$  can be made arbitrarily small [ES94, AB98, BO05].

Let us now consider a multi-material 3D image  $I$  as a function  $F : \mathbb{Z}^3 \rightarrow J$ , where  $J = \{0, \dots, n\}$  is a set of labels. Usually the label 0 represents the background and  $1 \dots n$  the other materials. Each label  $i$  defines a characteristic function  $f_i : \mathbb{Z}^3 \rightarrow \{0, 1\}$  such that  $f_i(p) = 1$  iff  $F(p) = i$  and  $f_i(p) = 0$  otherwise. Let  $\tilde{f}_i : \mathbb{R}^3 \rightarrow \{0, 1\}$  be the trilinear interpolation of  $f_i$ . Then we define the extension  $\tilde{F} : \mathbb{R}^3 \rightarrow J$  of the image function  $F$ , as follows:  $\tilde{F}(p) = i$  iff  $\tilde{f}_i(p) = \max_{j \in J} \{\tilde{f}_j(p)\}$ .  $\tilde{F}$  defines a partition of the domain to be meshed  $\Omega = \cup_{i \neq 0} \Omega_i$ , where  $\Omega_i = \tilde{F}^{-1}(i)$ ,  $i \in J$ . We call  $B_i$  the boundary of  $\Omega_i$  and denote  $B = \cup B_i$  the locus of multi-material junctions including surface patches, edges and corners.

The Delaunay refinement meshing algorithm presented in the next section samples a set of points  $E \in \mathbb{R}^3$  and builds the Delaunay triangulation of  $E$  restricted to  $\Omega$ . The algorithm outputs  $Del(E)|\Omega = \cup_{i \in J, i \neq 0} Del(E)|\Omega_i$  where  $Del(E)|\Omega_i$  is the set of tetrahedra whose circumcentres are contained in  $\Omega_i$ . Under the assumption that  $E$  is a sufficiently dense sample of  $B$ , each material of the image  $\Omega_i$  is represented by a submesh of tetrahedra, denoted  $Del(E)|\Omega_i$ . This submesh is homeomorphic to  $\Omega_i$  and its boundary is an accurate and homeomorphic surface approximation of the material boundary  $B_i$ . We call *boundary facets* the facets incident to two tetrahedra with different labels. These boundary facets form a watertight non-manifold surface mesh that approximates all multi-material junctions represented by the labelled input image.

Let us finally recall that every *boundary facet* has a dual Voronoi edge whose endpoints belong to different materials. Therefore, this Voronoi edge has a non empty intersection with some boundary surface  $B_i$ . Any point of this intersection is the centre of a ball circumscribing the facet and enclosing no sample points. Hereafter, such a ball is called a *surface Delaunay ball* of a boundary facet. Note that every boundary facet has at least one surface Delaunay ball because the intersection between its dual Voronoi edge and the surface always exists, by construction. Note that the surface Delaunay ball of a given boundary facet is usually unique, except in the first steps of the algorithm. However, even if there are several surface Delaunay ball centres any of them can be chosen, without compromising the good continuation of the algorithm.

## 2.2. Delaunay refinement algorithm

The algorithm starts by sampling a small initial set of points  $E$  on  $B$ . Three points per connected component of  $B$  are sufficient. Next it calculates the Delaunay triangulation  $Del(E)$ , its restriction  $Del(E)|\Omega$  and the boundary facets. This initial approximation is then refined until it meets the follow-

ing topological and quality criteria for boundary facets and tetrahedra.

### Refinement criteria for boundary facets:

- topological criterion – This criterion ensures that all vertices of a boundary facet lie on  $B$ ;
- minimum angle  $\alpha$  – This quality criterion ensures that boundary facets are well-shaped. The parameter  $\alpha$  is the lower bound for the minimum angle of any boundary facet. The algorithm is guaranteed to end if  $\alpha \leq 30^\circ$ .
- maximum edge length  $l$  – This sizing criterion is the upper bound for the maximum edge length of boundary facets. When the input of the algorithm is a 3D segmented image, this parameter is bounded from below by the image resolution. In practice, it has to be sufficiently large compared to the image resolution to ensure that the output mesh has a reasonable size.
- maximum surface distance  $d$  – This criterion controls the approximation accuracy of the mesh. The parameter  $d$  is the upper bound for the distance between the centre of a boundary facet and the centre of any surface Delaunay ball of this facet. This parameter is also bounded from below by the image resolution, but it has to be small compared to the previous maximum edge length parameter  $l$  to increase the approximation accuracy of the resulting mesh.

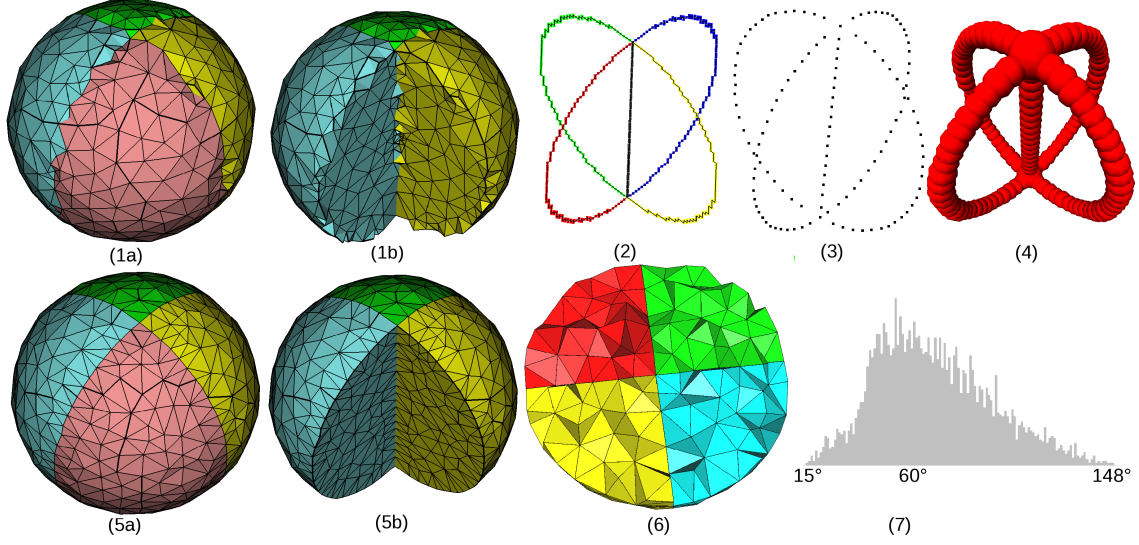
### Refinement criteria for tetrahedra:

- maximum radius-edge ratio  $\beta$  – This quality criterion ensures that a tetrahedron is well-shaped. The parameter is an upper bound for the ratio between a tetrahedron circumradius and its shortest edge length. The algorithm termination is guaranteed for  $\beta \geq 2$ , but our experimental results have shown that  $\beta = 4$  is a good compromise in the case of 3D images;
- maximum edge length  $L$  – This sizing criterion is the upper bound for the maximum edge length of any tetrahedron. This parameter has to be at least equal to the sizing criterion for boundary facets,  $L \geq l$ .

A *bad element* is a tetrahedron or a boundary facet that does not fulfil all the criteria. *Bad facets* are removed from the mesh by inserting the centres of their surface Delaunay balls (using the routine *refine\_facet*). *Bad tetrahedra* are refined by inserting their circumcentres (using the routine *refine\_tet*).

The algorithm is greedy. It inserts refinement points one by one and maintains the current set  $E$ ,  $Del(E)$ ,  $Del(E)|\Omega$  and boundary facets. It also maintains a list of *bad facets* and a list of *bad tetrahedra*. The algorithm combines two refinement rules and processes in priority boundary facets. The refinement routine is iterated until there is no bad element left as follows:

1. If there is a bad boundary facet  $f$  then *refine\_facet*( $f$ )
2. If there is a bad tetrahedron  $t$  in  $Del(E)|\Omega$



**Figure 2:** (1) Delaunay refinement 3D mesh. (2) Multi-material junctions: five 1-junctions and two 0-junctions. (3) Sampled points on junctions. (4) Protecting balls. (5) Feature preserving Delaunay refinement 3D mesh. (6) A cut of the tetrahedral mesh. (7) Histogram of the dihedral angles.

- a. compute the centre  $c$  of its circumscribing ball
- b. if  $c$  is included in a surface Delaunay ball of some boundary facet  $f$  then *refine\_facet*( $f$ )
- c. else *refine\_tet*( $t$ )

Delaunay refinement techniques are known to eliminate all badly shaped elements except one class of tetrahedra called *slivers* which have good radius-edge ratios but small volumes. We remove these almost flat tetrahedra from the final mesh using a *sliver exudation* algorithm [CDE\*99].

It is proven in [ORY05] that for appropriate choices of refinement criteria, the algorithm terminates. In practice, these criteria are given by the 5-uple of parameters  $(\alpha, l, d, \beta, L)$  which can be tuned to meet some user-given sizing requirements. The algorithm outputs surface and volume meshes which form a good approximation of the image partition as soon as  $E$  is a sufficiently dense sample of its boundaries and volumes. By construction,  $Del(E)|\Omega$  induces watertight surface meshes of each image material, free of self-intersections. These meshes are consistent at all multi-material junctions. However, since the 0- and 1-junctions are not handled explicitly, they are poorly represented in the output mesh. As shown on Fig.2(1), the 1-junction edges are usually zigzagging and their 0-junction intersection vertices are rarely preserved and may be multiple.

### 3. Feature preserving extension

In order to constrain the Delaunay refinement algorithm to mesh properly 0- and 1-junctions, firstly we need to extract

these junctions from the input 3D image which is, let us recall, the only available input data for our meshing method.

In this work, we propose an algorithm which extracts the 0- and 1-junctions from the *digital* subdivision of the domain to be meshed defined by the 3D image. The *digital* subdivision, borrowed from digital geometry and topology [Lat97, FB00], is slightly different from the trilinear subdivision induced by  $\tilde{F}$ , but they are very close. So, as it will be clarified hereafter, junctions defined into the digital subdivision can be successfully used to constrain junctions into the trilinear subdivision.

#### 3.1. Multi-material junction extraction

For the purposes of our novel multi-material junction extraction algorithm, we extend the image function  $F : \mathbb{Z}^3 \rightarrow J$  into a function  $G : \mathbb{R}^3 \rightarrow J$  such as for any point  $p \in \mathbb{R}^3$ ,  $G(p) = F(p_i)$ , where  $p_i$  is the point of  $\mathbb{Z}^3$  closest to  $p$ . As before, this function  $G$  defines a partition of the domain to be meshed  $\Omega = \cup_{i \neq 0} \Omega_i$ . But now  $\Omega_i = G^{-1}(i)$  for  $i \in J$ , is a set of face-connected unit cubes with the same label. These cubes are centred at points in  $\mathbb{Z}^3$  and have closed facets in  $\mathbb{R}^3$  parallel to one of the coordinate plans. We call *voxels* (3-cells) these cubes, *surfels* (2-cells) the unit facets bounding voxels, *linels* (1-cells) the unit segments bounding surfels and *pointels* (0-cells) the meeting points of linels. Every image region  $\Omega_i$  is represented by a 3D cellular complex which may contain more than one connected component. The intervoxel boundary of an image region  $\Omega_i$ , denoted  $B_i$ , is defined as the union of surfels which are incident to exactly one voxel in  $\Omega_i$ . The union of intervoxel boundaries, denoted



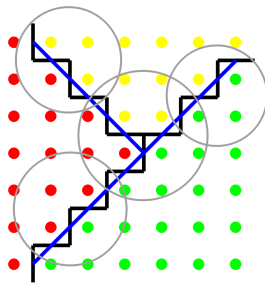
$B = \cup B_i$ , is typically a 2D cellular complex which describes all multi-material junctions between image regions, as follows:

- 2-junctions  $\{S_i\}$  are surface patches composed of connected surfels between exactly 2 materials;
- 1-junctions  $\{L_j\}$  are digital edges composed of connected set of linels separating 3 or more 2-junctions;
- 0-junctions  $\{P_k\}$  are corner pointels which are defined at meeting points of 3 or more 1-junctions.

As it has been stressed before, we want to constrain only 0- and 1-junctions because the Delaunay refinement algorithm handles surface patches well. Therefore, our junction extraction algorithm makes a single scan of the input 3D image and delivers a 1D cellular complex composed of digital edges  $\{L_j\}$  and their endpoints  $\{P_k\}$ . Each digital edge may be either a closed curve with no endpoint or a an edge with two endpoints. Fig.2(2) shows the 1D complex obtained for the multi-material sphere which is composed of five edges and two corners.

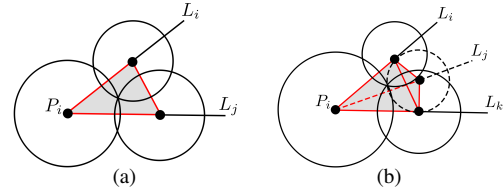
### 3.2. Protecting multi-material junctions

It is well known that neighbourhoods of 0- and 1-junctions are regions of potential problems for Delaunay refinement algorithms. Firstly, as we have already seen, the usual Delaunay refinement does not reconstruct these features accurately. Secondly, if we constrain these junctions to appear in the mesh and if the surface patches incident to them make dihedral angles smaller than  $90^\circ$ , the usual Delaunay refinement may not terminate. Recently, Cheng et al. [CDR07] have proposed a way to overcome this difficulty using *protecting balls* around the constrained edges and corners. In this work, we apply this mechanism to protect previously extracted 0- and 1-junctions with balls before launching the mesh refinement.



**Figure 3:** A 2D example of 0- and 1-junctions defined in the digital and the trilinear image subdivisions. The image data is represented by coloured dots. Corners and edges induced by functions  $G$  and  $\tilde{F}$  are respectively shown with black and blue thick lines. The difference between multi-material junctions defined in both subdivisions is contained in the union of the protecting balls.

Note that we use multi-material junctions, obtained from the subdivision induced by the function  $G$ , to constrain these junctions in the subdivision induced by the function  $\tilde{F}$ . Indeed, as shown on Figure 3, even if the 0- and 1-junctions defined by both subdivisions are very close they are not exactly the same. However, this difference is 'hidden' for the algorithm by the protecting balls. Indeed, if the radius of the protecting balls is large enough, the edges and the corners of both subdivisions are completely contained in the union of the protecting balls of these features.



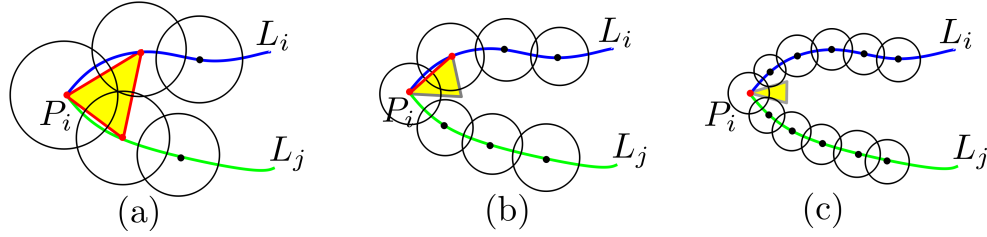
**Figure 4:** (a) Permitted configuration of 3 protecting balls with a common intersection. (b) Permitted configuration of 4 protecting balls with a common intersection.

For the protection step, we propose an hierarchical junction protection algorithm which first keeps all corner vertices in  $\{P_i\}$  and then samples points on edges  $\{L_i\}$  according to some user-specified distance  $d$  (see Fig.2(3)). In practice, this distance should be at most the maximum edge length parameter for facets  $l$ . Then the algorithm protects every sampled point  $p \in L_i$  with a ball  $b = (p, r)$  where  $r = 2/3 \times d$  (see Fig.2(4)). In our algorithm, the protecting balls have to satisfy the following properties:

- every edge  $L_i$  is completely covered by the protecting balls of its samples;
- any two adjacent balls on a given edge  $L_i$  overlap significantly without containing each other's centre;
- any two balls centred on different edges  $L_i$  and  $L_j$  do not intersect except if they both intersect a third ball, centred at the common 0-junction of  $L_i$  and  $L_j$  (see Fig.4(a));
- no three balls have a common intersection except if they are three consecutive balls on the same edge (which may happen when the edge is highly curved) or if they are three balls centred on  $L_i, L_j$  and  $L_i \cap L_j$  as in the previous case;
- no four balls have a common intersection except if three of them are centred on different edges  $L_i, L_j$  and  $L_k$  and if they intersect a fourth ball which is centred at  $L_i \cap L_j \cap L_k$  (see Fig.4(b)).

After the sampling and protection step, we enhance the previous Delaunay refinement algorithm in order to constrain the 0- and 1-junctions.

Each protecting ball  $b = (p, r)$  is turned into a weighted point  $(p, r)$  and inserted into the initial set of points  $E$ . The Delaunay triangulation is turned into a *weighted* Delaunay triangulation where the Euclidean distance is replaced by



**Figure 5:** (a) Boundary facet with 3 initial vertices whose protecting balls intersect. This facet is constrained to appear in the final triangulation without refinement. (b) Boundary facet with 2 initial vertices whose protecting balls intersect. The red Delaunay edge is constrained to appear in the final triangulation. The facet is refined only if it does not meet the topological criterion or is too big compared to the sizing criterion. (c) Boundary facet with one initial vertex which is refined only if it does not meet the topological or the sizing criteria for boundary facets.

the *weighted distance*. The weighted distance from a point  $x \in \mathbb{R}^3$  to a weighted point  $(p, r)$  is defined as  $\|x - p\|^2 - r^2$ . Accordingly, refinement points such as circumcentres of tetrahedra and surface Delaunay ball centres for boundary facets are replaced by their weighted counterparts.

Any vertex inserted by the refinement algorithm, after the protection step, has a zero weight. This enforces the fact two sampled points that are *consecutive* on a protected edge  $L_i$  are guaranteed to remain connected by a restricted Delaunay edge in the final mesh. Moreover, as it will be clarified hereafter, the algorithm never refines a boundary facet whose vertices correspond to three protecting balls with a non empty intersection, nor a tetrahedron whose vertices correspond to four intersecting balls. As a consequence, the refinement points are always inserted outside the union of the protecting balls.

### 3.3. Quality criteria adaptation

We describe here how to relax quality criteria for boundary facets and tetrahedra which are incident to the constrained 0- and 1-junctions in order to avoid infinite loops during the refinement. Indeed, such an element has at least one initial vertex which has been inserted during the protection step and its shape is more or less constrained by these initial points. Figure 5 gives an example of 3 types of boundary facets incident to 0- and 1-junctions.

Let us first consider the case of boundary facets incident to the protected junctions. To begin with, note that the refinement parameter  $l$ , which is the maximum edge length for a boundary facet, has to be at least as large as the length of the constrained edge on the 1-junction, otherwise there is no hope to be able to satisfy this sizing criterion. Then, for such a boundary facet, the algorithm considers only the topological and the sizing criteria and does not try to enforce its shape (minimum angle criterion  $\alpha$ ) or the approximation distance accuracy (maximum surface distance criterion  $d$ ). Indeed, because refinement points are always inserted outside the protecting ball, it may not be possible to satisfy shape

or distance criteria for a boundary facet with a vertex or an edge on a 1-junction. Trying to enforce one of these criteria would produce an endless loop in the refinement algorithm.

We use the following refinement criteria for a boundary facet  $f$  incident to multi-material junctions:

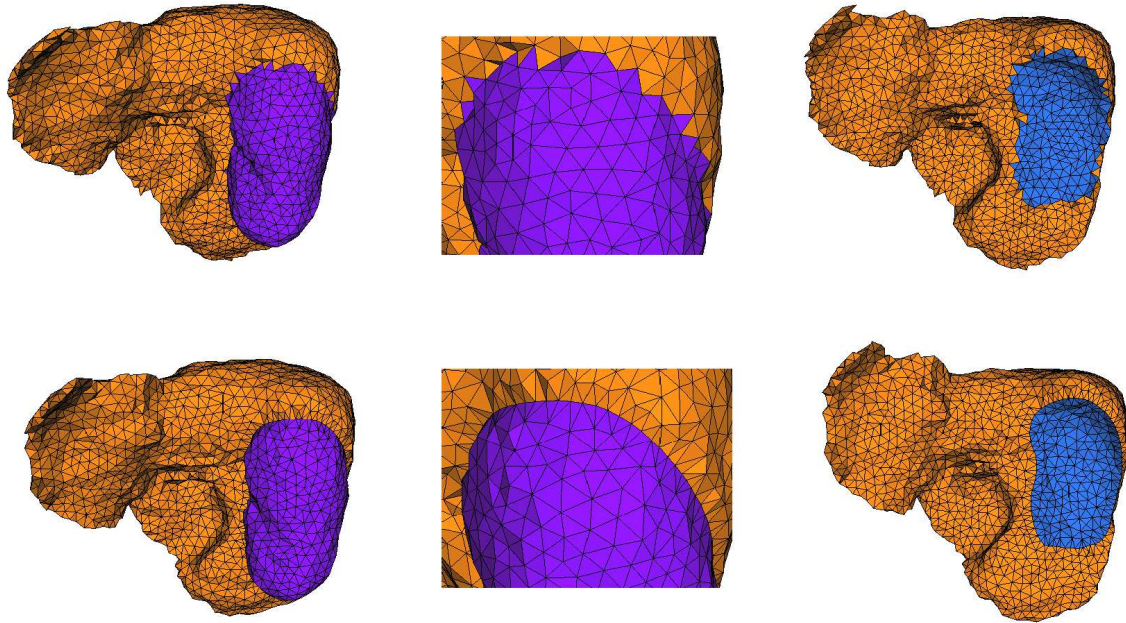
- If  $f$  has 3 initial vertices whose protecting balls intersect, the algorithm does not consider refinement criteria for this facet and the facet will appear in the final mesh;
- If  $f$  has 1 or 2 initial vertices whose protecting balls intersect, the algorithm considers only the topological criterion and the maximum edge length criterion  $l$  for this facet.

We also relax quality criteria for tetrahedra and use the following refinement criteria for a tetrahedron  $t$  incident to multi-material junctions:

- If  $t$  has 4 initial vertices whose protecting balls intersect, the tetrahedron is not refined and appears in the final mesh;
- If  $t$  has 3 initial vertices whose protecting balls intersect, the algorithm considers only the maximum edge length criterion  $L$  for this tetrahedron. However, this tetrahedron is refined only if its refinement point is not included in the surface Delaunay ball of the boundary facet defined by the 3 initial vertices. Otherwise, the tetrahedron is definitely accepted;
- If  $t$  has 1 or 2 weighted vertices, the algorithm considers only the maximum edge length criterion  $L$  for this tetrahedron.

As for boundary facets, the constrained Delaunay refinement algorithm does not consider the maximum radius-edge ratio criterion  $\beta$  for these 4 types of tetrahedra to avoid infinite loops if these elements do not meet the criterion.

Note that dropping few quality criteria for boundary facets and tetrahedra that are constrained to 0- and 1-junctions may slightly decrease the overall mesh quality. However, our experimental results tend to show that this small alteration is imperceptible compared to the significant improvement of the mesh quality along these features.



**Figure 6:** Meshes generated from a segmented medical image where a liver is adjacent to a right kidney. The first row shows meshes obtained with the usual Delaunay refinement. The second row shows meshes resulting from our feature preserving extension. The 3rd column shows the common interface between both organs.

The protecting balls and the weighted Delaunay triangulation guarantee two important properties.

First, any two sampled points that are *consecutive* on a protected edge  $L_i$  are guaranteed to remain connected by a restricted Delaunay edge in the final mesh. Note that the size of this mesh edge is defined by the chosen distance  $d$  between sampled points and the resulting protection radius. In practice, as it has been explained before, this parameter should be large enough so that the union of the protecting balls covers the difference between the digital and the trilinear image subdivisions. In our current implementation, this parameter is either user-given or set to the maximum edge length parameter for boundary facets  $l$ .

Second, the termination of the algorithm is guaranteed because it never tries to insert a point into the union of protecting balls and it uses relaxed quality criteria for Delaunay simplices in the proximity of the constrained junctions.

In practice, when the input is a multi-material 3D image, the algorithm outputs high-quality tetrahedral meshes of the different materials that have consistent multi-material junctions. Moreover, the 0-junctions are preserved and the 1-junctions are correctly approximated by edges whose vertices lie on these junctions (see Fig.2(5)).

#### 4. Experimental results

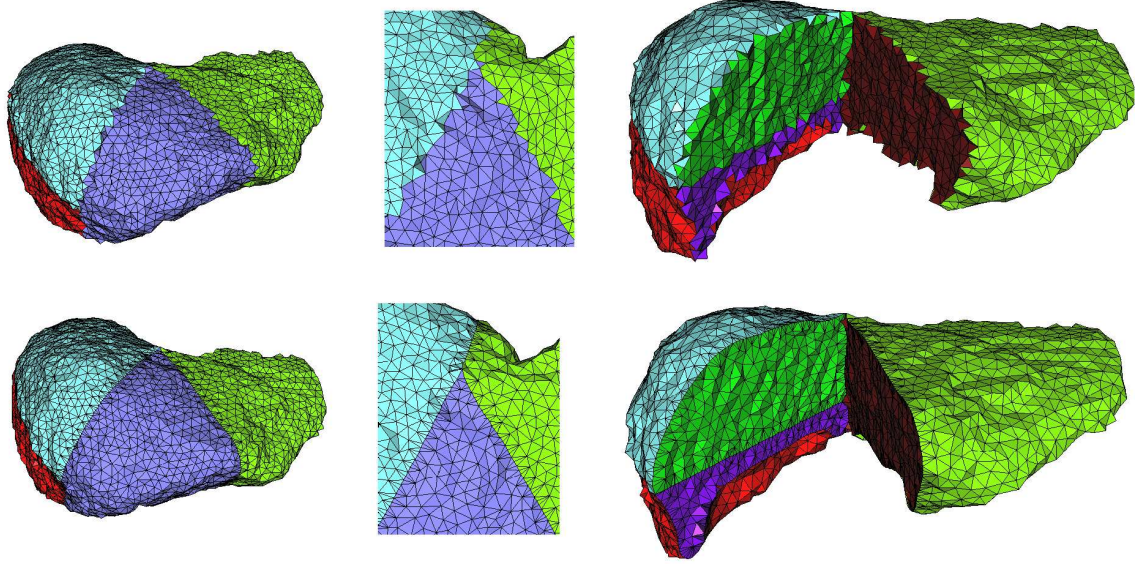
The Delaunay refinement algorithm and its feature preserving extension have been implemented using robust primitives to compute the Delaunay triangulation provided by the CGAL library [CGA].

We have tested our meshing algorithm on synthetic labelled images and on various segmented medical images which have been provided by IRCAD [IRC]. Figures 6 and 7 shows the meshes generated from two segmented liver datasets by the Delaunay refinement strategy with and without its feature preserving extension. Note that these meshes have been generated from segmented images without any time-consuming preprocessings of the input data. However, if the target application requires smoother geometric representations, the resulting meshes can be optimized using the method presented in [ACSYD05, TWAD09].

Figure 6 displays the mesh of a liver adjacent to a right kidney and their 1-junction edge defined at the intersection of these two objects and the background. In this case, the 1-junction is a simple closed curve. We have chosen to protect this curve with balls of radius  $6mm$  and to mesh the organs with maximum edge length for boundary facets  $l = 12mm$ .

Figure 7 shows the mesh of a liver which has been previously segmented into 4 anatomical regions using cutting planes. There are 8 1-junctions and 4 0-junctions defined at the intersection of these anatomical regions. We have protected these features with balls of radius  $3mm$  and have





**Figure 7:** Meshes generated from a segmented liver image representing 4 anatomical liver regions. The first row shows meshes obtained with the usual Delaunay refinement algorithm. The second row shows meshes generated with our feature preserving extension. The 3rd column shows some internal interfaces between the anatomical regions.

Experiment	sphere	liver-kidney	liver segments
Image size	$62 \times 62 \times 62$	$512 \times 512 \times 112$	$402 \times 356 \times 238$
Image resolution (mm)	$1 \times 1 \times 1$	$0.67 \times 0.67 \times 2$	$2 \times 2 \times 2$
Refinement criteria ( $\alpha, l, d, \beta, L$ )	(20,10,3,4,10)	(30,12,2,4,14)	(25,14,4,4,18)
# vertices	964	6142	12381
# boundary facets	1431	5439	9646
# tetrahedra	4434	31043	64485
Junction Extraction (sec)	0.72	4.56	21.35
Surface meshing (sec)	1.74	9.99	11.04
Volume meshing (sec)	1.13	5.82	17.23
Sliver exudation (sec)	3.75	13.82	48.64

**Table 1:** Quantitative results and parameters for three different 3D images. The  $\alpha$  in the refinement criteria is given in degree and  $l, d$  and  $L$  are given in mm. The four last rows give the computation times of different algorithm steps in seconds.

meshed the regions with maximum edge length for boundary facets  $l = 16\text{mm}$ .

Table 1 lists the quantitative results for these two liver images and the multi-material sphere on Fig.2. The refinement criteria for Delaunay refinement are given as the 5-uplet  $(\alpha, l, d, \beta, L)$  defined in Section 2. Note that our edge extraction and protection algorithm takes a few seconds while the Delaunay refinement takes about 3 times more. In practice, the algorithm has a reasonable computation time. A typical liver image ( $512 \times 512 \times 112$ ) segmented into 20 different materials is usually meshed with target edge length of  $10\text{mm}$  in less than half a minute.

## 5. Conclusion

In this paper, we have proposed an efficient and robust feature preserving Delaunay refinement meshing strategy for segmented 3D images. Despite the topological complexity of multi-material junctions, our algorithm delivers high-quality meshes of image regions which are consistent with each other along their common junctions. The method first extracts, from the input 3D image, all multi-material junctions where 3 or more regions meet and uses a constrained Delaunay refinement algorithm which allows to approximate accurately these sharp features in the output mesh.

The presented algorithm was successfully applied to gen-

erate meshes from segmented MRI and CT-scan images which have been used for visualization of organs for diagnosis and education. The resulting meshes are also targeted towards applications in finite element methods which require conforming multi-material junctions to avoid instabilities and errors during the simulation [SCL\*06,CLD\*07].

As previously mentioned, one of our future works will concern mesh optimization based on the method presented in [TWAD09]. We also plan to make our code available very soon and we hope that it will be useful in fields which require realistic geometric models.

## 6. Acknowledgments

The authors acknowledge the support of the FOCUS K3D Coordination Action, EU Contract ICT-2007.4.2 contract number 214993. We would also like to thank the anonymous reviewers for their valuable comments. All of the segmented medical images in this paper are courtesy of IRCAD.

## References

- [AB98] AMENTA N., BERN M.: Surface reconstruction by voronoi filtering. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry* (NY, USA, 1998), ACM, pp. 39–48.
- [ACSYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics* 24 (2005), 617–625.
- [BH05] BERTRAM M., HAGEN H.: Non-manifold mesh extraction from time-varying segmented volumes used for modeling a human heart. In *EuroVis* (2005), pp. 199–206.
- [BK06] BISCHOFF S., KOBELT L.: Extracting consistent and manifold interfaces from multi-valued volume data sets. In *Bildverarbeitung für die Medizin* (2006).
- [BO05] BOISSONNAT J.-D., OUDOT S.: Provably good sampling and meshing of surfaces. *Graphical Models* 67 (2005), 405–451.
- [BY98] BOISSONNAT J.-D., YVINEC M.: *Algorithmic geometry*. Cambridge University Press, UK, 1998.
- [CDE\*99] CHENG S.-W., DEY T. K., EDELSBRUNNER H., FACELLO M. A., TENG S.-H.: Sliver exudation. In *SCG* (1999), pp. 1–13.
- [CDL07] CHENG S.-W., DEY T. K., LEVINE J. A.: A practical delaunay meshing algorithm for a large class of domains. In *Meshing Roundtable* (2007), pp. 477–494.
- [CDR07] CHENG S.-W., DEY T. K., RAMOS E. A.: Delaunay refinement for piecewise smooth complexes. In *SODA* (Philadelphia, PA, USA, 2007), pp. 1096–1105.
- [CGA] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [CLD\*07] CLATZ O., LITRICO S., DELINGETTE H., PAQUIS P., AYACHE N.: Dynamic model of communicating hydrocephalus for surgery simulation. *IEEE Transactions on Biomedical Engineering* 54, 4 (April 2007), 755–758.
- [ES94] EDELSBRUNNER H., SHAH N. R.: Triangulating topological spaces. In *SCG '94: Proceedings of the tenth annual symposium on Computational geometry* (NY, USA, 1994), ACM, pp. 285–292.
- [FB00] FRANÇON J., BERTRAND Y.: Topological 3d-manifolds: a statistical study of the cells. *Theoretical Computer Science* (2000), 233–254.
- [Gib98] GIBSON S. F. F.: Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In *MICCAI* (London, UK, 1998), Springer-Verlag, pp. 888–898.
- [HK98] HARTMANN U., KRUGGEL F.: A fast algorithm for generating large tetrahedral 3d finite element meshes from magnetic resonance tomograms. In *WBIA* (1998).
- [HSSZ97] HEGE H.-C., SEEBAB M., STALLING D., ZOCKLER M.: *A generalized marching cubes algorithm based on non-binary classification*. Tech. rep., 1997.
- [IRC] IRCAD, Institut de Recherche contre les Cancers de l'Appareil Digestif. <http://www.ircad.fr>.
- [Lat97] LATECKI L.: 3d well-composed pictures. *Graph. Models Image Process.* 59, 3 (1997).
- [MWK\*08] MEYER M., WHITAKER R., KIRBY R. M., LEDERGERBER C., PFISTER H.: Particle-based sampling and meshing of surfaces in multimaterial volumes. *Visualization and Computer Graphics* 14, 6 (2008), 1539–1546.
- [NF97] NIELSON G. M., FRANKE R.: Computing the separating surface for segmented data. In *VIS* (1997), pp. 229–233.
- [Nie04] NIELSON G. M.: Dual marching cubes. In *VIS* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 489–496.
- [ORY05] OUDOT S., RINEAU L., YVINEC M.: Meshing volumes bounded by smooth surfaces. In *Meshing Roundtable* (2005), pp. 203–219.
- [PSB\*07] PONS J.-P., SÉGONNE F., BOISSONNAT J.-D., RINEAU L., YVINEC M., KERIVEN R.: High-quality consistent meshing of multi-label datasets. In *IPMI* (2007), pp. 198–210.
- [RBB05] REITINGER B., BORNIK A., BEICHEL R.: Constructing smooth non-manifold meshes of multi-labeled volumetric datasets. In *WSCG* (2005), pp. 227–234.
- [RY07] RINEAU L., YVINEC M.: Meshing 3d domains bounded by piecewise smooth surfaces. In *Meshing Roundtable* (2007), pp. 443–460.
- [SCL\*06] SCARELLA G., CLATZ O., LANteri S., BEAUME G., OUDOT S., PONS J.-P., PIPERNO S., JOLY P., WIART J.: Realistic numerical modelling of human head tissue exposure to electromagnetic waves from cellular phones. *Comptes Rendus de l'Académie des Sciences - Physics* 7, 5 (June 2006), 501–508.
- [She00] SHEWCHUK J. R.: Mesh generation for domains with small angles. In *SCG* (New York, USA, 2000), ACM, pp. 1–10.
- [TWAD09] TOURNOIS J., WORMSER C., ALLIEZ P., DESBRUN M.: Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. *ACM Transactions on Graphics* 28(3) (2009).
- [WS03] WU Z., SULLIVAN J. M.: Multiple material marching cubes algorithm. *Int. J. Numer. Methods Eng.* (2003), 189–207.
- [ZHB07] ZHANG Y., HUGHES T., BAJAJ C. L.: Automatic 3d mesh generation for a domain with multiple materials. In *Meshing Roundtable* (2007), pp. 367–386.